FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 30, N° 1, March 2017, pp. 49 - 66
DOI: 10.2298/FUEE1701049A

AN EXTENDED GREEN - SASAO HIERARCHY OF CANONICAL TERNARY GALOIS FORMS AND UNIVERSAL LOGIC MODULES

Anas N. Al-Rabadi[1,2]

[1]Electrical Engineering Department, Philadelphia University,
[2]Jordan & Computer Engineering Department,
The University of Jordan, Amman-Jordan

Abstract. *A new extended Green-Sasao hierarchy of families and forms with a new subfamily for multiple-valued Reed-Muller logic is introduced. Recently, two families of binary canonical Reed-Muller forms, called Inclusive Forms (IFs) and Generalized Inclusive Forms (GIFs) have been proposed, where the second family was the first to include all minimum Exclusive Sum-Of-Products (ESOPs). In this paper, we propose, analogously to the binary case, two general families of canonical ternary Reed-Muller forms, called Ternary Inclusive Forms (TIFs) and their generalization of Ternary Generalized Inclusive Forms (TGIFs), where the second family includes minimum Galois Field Sum-Of-Products (GFSOPs) over ternary Galois field GF(3). One of the basic motivations in this work is the application of these TIFs and TGIFs to find the minimum GFSOP for multiple-valued input-output functions within logic synthesis, where a GFSOP minimizer based on IF polarity can be used to minimize the multiple-valued GFSOP expression for any given function. The realization of the presented Shannon-Davio (S/D) trees using Universal Logic Modules (ULMs) is also introduced, where ULMs are complete systems that can implement all possible logic functions utilizing the corresponding S/D expansions of multiple-valued Shannon and Davio spectral transforms.*

Key words: *Canonical Forms, Galois Field Forms, Green-Sasao Hierarchy, Inclusive Forms, Multiple-Valued Logic, Shannon-Davio Trees, Ternary Logic, Universal Logic Modules.*

1 Normal Galois Forms

Reed-Muller-like spectral transforms [1-18] have found a variety of useful applications in minimizing Exclusive Sum-Of-Products (ESOP) and Galois field SOP (GFSOP) expressions, creation of new forms, binary and spectral decision diagrams, regular structures, besides the well-known uses in digital communications, digital signal and image processing, and fault detection and testing [1-9, 12-14, 16, 17, 19, 20, 21]. The method of generating the new families of multiple-valued Shannon and Davio spectral transforms is based on the fundamental multiple-valued Shannon and Davio expansions. Dyadic families of discrete transforms; Reed-Muller and Green-Sasao hierarchy, Walsh, Arithmetic, Adding and Haar transforms and their generalizations to multiple-valued transforms, have also found important applications in digital system design and optimization [1, 6, 19, 7-18, 20-31].

Received November 28, 2015; received in revised form April 13, 2016
Corresponding author: Anas N. Al-Rabadi
Electrical Engineering Department, Philadelphia University, Jordan & Computer Engineering Department,
The University of Jordan, Amman-Jordan
(email: alrabadi@yahoo.com)

Table 1: Number of product terms needed to realize some arithmetic functions using various expressions.

Function	PPRM	FPRM	GRM	ESOP	SOP
adr4	34	34	34	31	75
log8	253	193	105	96	123
nrm4	216	185	96	69	120
rdm8	56	56	31	31	76
rot8	225	118	51	35	57
sym9	210	173	126	51	84
wgt8	107	107	107	58	255

Normal canonical forms play an important role in the synthesis of logic circuits which includes synthesis, testing and optimization [1, 9, 12-15, 17, 21, 22, 26, 27, 31, 32]. One can observe that by going, for example, from Positive Polarity Reed-Muller (PPRM) form to the Generalized Reed-Muller (GRM) form, less constraints are imposed on the canonical forms due to the enlarged set of polarities that one can choose from. The gain of more freedom on the polarity of the canonical expansions will provide an advantage of obtaining Exclusive-Sum-Of-Product (ESOP) expressions with less number of terms and literals, and consequently expressing Boolean functions using ESOP forms will produce on average expressions with less size as if compared to Sum-Of-Product (SOP) expressions for example. Table 1 illustrates these observations [1]. The main algebraic structure which is used in this work for developing the canonical normal forms is the Galois field (GF) algebraic structure, which is a fundamental algebraic structure in the theory of algebra [1, 12, 13, 17, 22, 29, 33, 34]. The importance of GF for logic synthesis results from the fact that every finite field is isomorphic to a Galois field. In general, the attractive properties of GF-based circuits, such as the high testability of such circuits, are mainly due to the fact that the GF operators exhibit the Cyclic Group, also known as Latin Square, property.

In binary, for example, GF(2) addition gate, the EXOR, has the cyclic group property. The cyclic group property can be explained, for example, using the three-valued (ternary) GF operators as shown in Figures 1(a) and 1(b), respectively. Note that in any row and column of the addition table in Figure 1(a), the elements are all different, which is cyclic, and that the elements have a different order in each row and column. Another cyclic group can be observed in the multiplication table; if the zero elements are removed from the multiplication table in Figure 1(b), then the remaining elements form a cyclic group.

Reed-Muller normal forms have been classified using the Green-Sasao hierarchy [1, 10, 12, 13, 17], where the Green-Sasao hierarchy of families of canonical forms and corresponding decision diagrams is based on three generic expansions; Shannon, positive Davio and negative Davio expansions. The two-valued Shannon, positive Davio and negative Davio expansions are given as follows, respectively:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

(a)

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

(b)

Figure 1: Third radix Galois field addition and multiplication tables: (a) addition and (b) multiplication.

$$\begin{aligned} f(x_1,x_2,\ldots,x_n) &= \bar{x}_1 \cdot f_0(x_1,x_2,\ldots,x_n) \oplus x_1 \cdot f_1(x_1,x_2,\ldots,x_n) \\ &= \begin{bmatrix} \bar{x}_1 & x_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}, \end{aligned} \tag{1}$$

$$\begin{aligned} f(x_1,x_2,\ldots,x_n) &= 1 \cdot f_0(x_1,x_2,\ldots,x_n) \oplus x_1 \cdot f_2(x_1,x_2,\ldots,x_n) \\ &= \begin{bmatrix} 1 & x_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}, \end{aligned} \tag{2}$$

$$\begin{aligned} f(x_1,x_2,\ldots,x_n) &= 1 \cdot f_1(x_1,x_2,\ldots,x_n) \oplus \bar{x}_1 \cdot f_2(x_1,x_2,\ldots,x_n) \\ &= \begin{bmatrix} 1 & \bar{x}_1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}, \end{aligned} \tag{3}$$

where $f_0(x_1,x_2,\ldots,x_n) = f(0,x_2,\ldots,x_n) = f_0$ is the negative cofactor of variable x_1, $f_1(x_1,x_2,\ldots,x_n) = f(1,x2,\ldots,x_n) = f_1$ is the positive cofactor of variable x_1, and $f_2(x_1,x_2,\ldots,x_n) = f(0,x_2,\ldots,x_n) \oplus f(1,x_2,\ldots,x_n) = f_0 \oplus f_1$. In addition, an arbitrary n-variable function $f(x_1,\ldots,x_n)$ can be represented using PPRM expansion as [1, 17]:

$$\begin{aligned} f(x_1,x_2,\ldots,x_n) =& a_0 \oplus a_1x_1 \oplus \ldots \oplus a_nx_n \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{n-1,n}x_{n-1}x_n \oplus \\ & \ldots \oplus a_{12\ldots n}x_1x_2\ldots x_n. \end{aligned} \tag{4}$$

For each function f, the coefficients a_i in Equation (4) are determined uniquely, so PPRM is a canonical form. For example, if we use either only the positive literal or only the negative literal for each variable in Equation (4) we obtain the Fixed Polarity Reed-Muller (FPRM) form.

The good selection of different permutations using Shannon and Davio expansions - like other expansions such as Walsh and Arithmetic expansions - as internal nodes in decision trees (DTs) and diagrams (DDs) will result in DTs and DDs that represent the corresponding logic functions with smaller sizes in terms of the total number of hierarchical levels used, and the total number of internal nodes needed.

In general, a literal can be defined as any function of a single variable. Basis functions in the general case of multiple-valued expansions are constructed using these

literals. Galois field Sum-Of-Products expansions can be performed utilizing variety uses of literals. For example, one can use K-Reduced Post literal (K-RPL) to produce K-RPL GFSOP, Generalized (Post) literal (GL) to produce GL GFSOP, and Universal literal (UL) to produce UL GFSOP.

Example 1. *Figure 2 demonstrates several literal types, where one proceeds from the simplest literal in Figure 2(a) (i.e., RPL) to the most complex universal literal in Figure 2(c). For RPL in Figure 2(a), a value K is produced by the literal when the value of the variable is equal to a specific state, and in this particular example a value of* $K = 1$ *is generated by the 1-RPL when the value of variable x is equal to certain state where this state here equals to one. The GL in Figure 2(b) produces a value of radix for a set of distinct states. One notes that, in contrast to the other literals, UL in Figure 2(c) can have any value of the logic system at distinct states, and thus UL has the highest complexity among the different types of literals.*

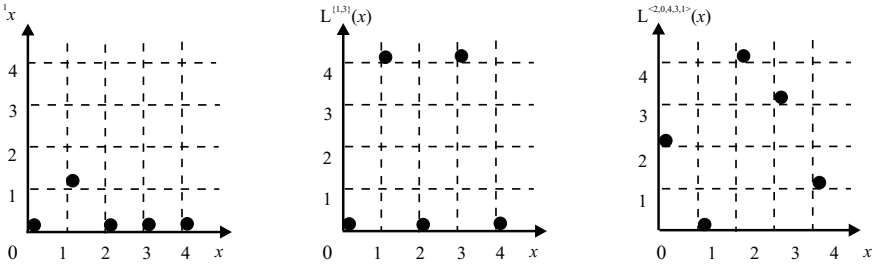


Figure 2: An illustrating example of the different types of literals over an arbitrary five-radix logic: (a) 1-Reduced Post Literal (1-RPL), (b) Generalized (Post) Literal (GL), and (c) Universal Literal (UL).

Since K-RPL GFSOP is simpler from implementation point of view than GL or UL, we will perform all the GFSOP expansions utilizing the 1-RPL GFSOP. Let us define 1-RPL [1, 17] as:

$$^ix = 1 \text{ iff } x = i \text{ else } {}^ix = 0. \tag{5}$$

For example $\{^0x,\ ^1x,\ ^2x\}$ are the zero, first, and second polarities of the 1-RPL, respectively. Also, let us define the ternary shifts over variable x as $\{x, x', x''\}$ as the zero, first and second shifts of the variable x respectively, i.e., $x = x + 0$, $x' = x + 1$ and $x'' = x + 2$, and x can take any value in the set $\{0, 1, 2\}$. We chose to represent the 1-Reduced Post Literals in terms of shifts and powers, among others, because of the ease of the implementation of powers of shifted variables in hardware as will be seen in Section 3 within Universal Logic Modules (ULMs) for the production of RPL. The fundamental Shannon expansion over GF(3) for a ternary function with a single variable is shown in Theorem 1.

Theorem 1. *Shannon expansion over GF(3) for a function with a single variable is:*

$$f = {}^0xf_0 + {}^1xf_1 + {}^2xf_2, \tag{6}$$

where f_0 *is cofactor of* f *with respect to variable* x *of value* 0*,* f_1 *is cofactor of* f *with respect to variable* x *of value* 1*, and* f_2 *is cofactor of* f *with respect to variable* x *of value 2.*

Proof. From Equation (5), if we substitute the values of the 1-RPL in Equation (6), we obtain the following $\{x = 0 \Rightarrow f_{x=0} = f_0, x = 1 \Rightarrow f_{x=1} = f_1, x = 2 \Rightarrow f_{x=2} = f_2\}$ which are the cofactors of variable x of values $\{0,1,2\}$, respectively. □

Example 2. *Let $f(x_1,x_2) = x_1' x_2 + x_2'' x_1$. Then, using Figure 1, the ternary truth vector with the variable order $\{x_1,x_2\}$ is $F = [0,2,1,1,2,0,2,2,2]^T$. Using Equation (6), we obtain the following GF(3) Shannon expansion $f(x_1,x_2) = {}^0x_1\ {}^1x_2 + 2\cdot\ {}^0x_1\ {}^2x_2 + 2\cdot\ {}^1x_1\ {}^0x_2 + 2\cdot\ {}^1x_1\ {}^1x_2 + 2\cdot\ {}^1x_1\ {}^2x_2 +\ {}^2x_1\ {}^0x_2 + 2\cdot\ {}^2x_1\ {}^2x_2$.*

By using the addition and multiplication over GF(3) utilizing Figure 1, the 1-RPL which is defined in Equation (5) is related to the shifts of variables over GF(3) in terms of powers as follows:

$$
\begin{aligned}
{}^0x &= 2(x)^2 + 1, && (7)\\
{}^0x &= 2(x')^2 + 2(x'), && (8)\\
{}^0x &= 2(x'')^2 + x'', && (9)\\
{}^1x &= 2(x)^2 + 2(x), && (10)\\
{}^1x &= 2(x')^2 + x', && (11)\\
{}^1x &= 2(x'')^2 + 1, && (12)\\
{}^2x &= 2(x)^2 + x, && (13)\\
{}^2x &= 2(x')^2 + 1, && (14)\\
{}^2x &= 2(x'')^2 + 2(x''). && (15)
\end{aligned}
$$

After the substitution of Equations (7) - (15) in Equation (6), and after the minimization of the terms according to the GF operations in Figure 1, one obtains the following Equations:

$$
\begin{aligned}
f &= 1\cdot f_0 + x\cdot(2f_1 + f_2) + 2(x)^2(f_0 + f_1 + f_2), && (16)\\
f &= 1\cdot f_2 + x'\cdot(2f_0 + f_1) + 2(x')^2(f_0 + f_1 + f_2), && (17)\\
f &= 1\cdot f_1 + x"\cdot(2f_2 + f_0) + 2(x'')^2(f_0 + f_1 + f_2). && (18)
\end{aligned}
$$

Equations (6) and (16) - (18) are the ternary fundamental Shannon and Davio expansions for a single variable, respectively. These Equations can be re-written in the following matrix-based forms as shown in Equations (19) - (22). We observe that Equations (19) - (22) are expansions for a single variable, but these expansions can be recursively generated for arbitrary number of variables N using the Kronecker product - also called the tensor product - analogously to the binary case [1, 17].

$$f = \begin{bmatrix} {}^0x & {}^1x & {}^2x \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}, \tag{19}$$

$$f = \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}, \tag{20}$$

$$f = \begin{bmatrix} 1 & x' & (x')^2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 2 & 1 & 0 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}, \tag{21}$$

$$f = \begin{bmatrix} 1 & x'' & (x'')^2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}. \tag{22}$$

The recursive generation using the Kronecker product for arbitrary number of variables can be expressed formally as in the following forms for ternary Shannon (S) and Davio $(\mathrm{D}_0, \mathrm{D}_1, \mathrm{D}_2)$ expansions, respectively:

$$f = \bigotimes_{i=1}^{N} \begin{bmatrix} {}^0x_i & {}^1x_i & {}^2x_i \end{bmatrix} \bigotimes_{i=1}^{N} [\mathrm{S}][\vec{F}], \tag{23}$$

$$f = \bigotimes_{i=1}^{N} \begin{bmatrix} 1 & x_i & x_i^2 \end{bmatrix} \bigotimes_{i=1}^{N} [\mathrm{D}_0][\vec{F}], \tag{24}$$

$$f = \bigotimes_{i=1}^{N} \begin{bmatrix} 1 & x_i' & (x_i')^2 \end{bmatrix} \bigotimes_{i=1}^{N} [\mathrm{D}_1][\vec{F}], \tag{25}$$

$$f = \bigotimes_{i=1}^{N} \begin{bmatrix} 1 & x_i'' & (x_i'')^2 \end{bmatrix} \bigotimes_{i=1}^{N} [\mathrm{D}_2][\vec{F}]. \tag{26}$$

Analogously to the binary case, we can have expansions that are mixed of Shannon (S) for certain variables and Davio $(\mathrm{D}_0, \mathrm{D}_1, \mathrm{D}_2)$ for the other variables. This will lead, analogously to the binary case, to the Kronecker Ternary Decision Trees (TDTs). Moreover, mixed expansions can be extended to include the case of Pseudo Kronecker TDTs [17].

2 New Multiple-Valued S/D Trees and Their Canonical Galois SOP Forms

Economical and highly testable implementations of Boolean functions, based on Reed-Muller (AND-EXOR) logic, play an important role in logic synthesis and circuit design. The AND-EXOR circuits include canonical forms which are expansions that are unique representations of a Boolean function. Several large families of canonical forms: Fixed polarity Reed-Muller (FPRM) forms, generalized Reed-Muller (GRM) forms, Kronecker (KRO) forms and pseudo-Kronecker (PSDKRO) forms, referred to

as the Green-Sasao hierarchy, have been described [1, 10, 17]. Because canonical families have higher testability and some other properties desirable for efficient synthesis, especially of some classes of functions, they are widely investigated. A similar ternary version of the binary Green-Sasao hierarchy can be developed, where this hierarchy can find applications in minimizing Galois field Sum-Of-Product (GFSOP) expressions which are expressions in the Sum-Of-Product form that uses the additions and multiplications of arbitrary radix Galois field, and can be used for the creation of new forms, decision diagrams and regular structures.

The current state-of-the-art minimizers of Exclusive Sum-Of-Product (ESOP) expressions are based on heuristics and give the exact solution only for functions with a small number of variables. For example, a formulation for finding exact ESOP was given [11], and an algorithm to derive minimum ESOP for 6-variable function was provided [25]. Because GFSOP minimization is even more difficult, it is important to investigate the structural properties and the counts of their canonical sub-families.

Two families of binary canonical Reed-Muller forms, called Inclusive Forms (IFs) and Generalized Inclusive Forms (GIFs) were presented [1], where the second family was the first to include all minimum ESOPs (binary GFSOPs). In these forms, IF is the form generated by the corresponding S/D tree and GIF is the form which is the union of the various variable-based ordering IFs (cf. definitions and theorems of these forms in the next subsection). In this paper, we propose, as analogous to the binary case, two general families of canonical ternary Reed-Muller forms, called Ternary Inclusive Forms (TIFs) and their generalization of Ternary Generalized Inclusive Forms (TGIFs), where GFSOP minimizer based on these new forms can be used to minimize functional GFSOP expressions and the second family of TGIFs includes minimum GFSOPs over ternary Galois field. One of the motivations for this work is the application of these TIFs and TGIFs to find minimum GFSOP for multiple-valued inputs multiple-valued outputs within logic synthesis, where the corresponding S/D trees provide more general polarity that contains GRM forms as a special case.

2.1 S/D Trees and their Inclusive Forms and Generalized Inclusive Forms

Two general families based on the Shannon expansion and the generalized Davio expansion which are produced using the corresponding S/D trees are presented in this subsection. These families are called the Inclusive Forms (IFs) and the Generalized Inclusive Forms (GIFs). The corresponding expansions over GF(2) are shown in Figure 3, where Figure 3(d) shows the new node which is based on binary Davio expansions called the generalized Davio (D) expansion (cf. Equation (28) for the more general ternary case) that generates the negative and positive Davio expansions as special cases.

The S/D trees for IFs of two variables can be generated for variable order $\{a,b\}$ and for variable order $\{b,a\}$ as well. The set of GIFs for two variables is the union of these two order-based IFs, where the total number of the resulting GIFs is equal to:

$$\#\text{GIF} = 2 \cdot (\#\text{IF}_{a,b}) - \#(\text{IF}_{a,b} \bigcap \text{IF}_{b,a}). \tag{27}$$

The Galois-based Shannon and Davio ternary expansions (i.e., flattened forms) can be represented in Ternary DTs (TDTs) and the corresponding varieties of Ternary DDs (TDDs) according to the corresponding reduction rules that are used [1, 17]. For one

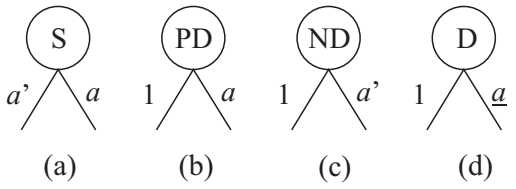


Figure 3: Two-valued nodes: (a) Shannon, (b) positive Davio, (c) negative Davio, and (d) generalized Davio where $\underline{a} \in \{a, a'\}$.

variable, that is one tree level, Figure 4 represents the expansion nodes for ternary Shannon, Davio and generalized Davio (D), respectively.

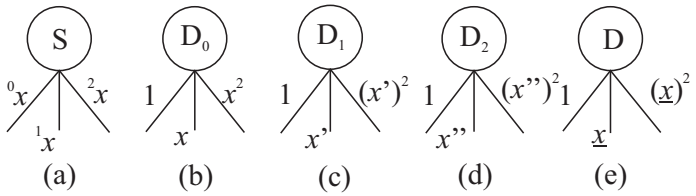


Figure 4: Ternary nodes for ternary decision trees: (a) Shannon, (b) Davio0, (c) Davio1, (d) Davio2, and (e) generalized Davio (D) where $\underline{x} \in \{x, x', x''\}$.

In correspondence to the binary S/D trees, one can produce ternary S/D trees. To define the ternary S/D trees, we define the generalized Davio node over ternary Galois radix as shown in Figure 4(e). Our notation here is that $(\underline{x})$ corresponds to the three possible shifts of the variable x as follows:

$$\underline{x} \in \{x, x', x''\}, \text{ over GF(3)}. \tag{28}$$

Definition 1. *The ternary tree with ternary Shannon and ternary generalized Davio nodes, that generates other ternary trees, is called ternary Shannon-Davio (S/D) tree.*

Utilizing the definition of ternary Shannon in Figure 4(a) and ternary generalized Davio in Figure 4(e), we obtain the ternary Shannon-Davio trees (ternary S/D trees) for two variables as shown in Figure 5. From the ternary S/D DTs shown in Figure 5, if we take any S/D tree and multiply the second-level cofactors (which are in the TDT leaves) each by the corresponding path in that TDT, and sum all the resulting cubes (terms) over GF(3), we obtain the flattened form of the function f, as a certain GFSOP expression. For each TDT in Figure 5, there are as many forms obtained for the function f as the number of possible permutations of the polarities of the variables in the second-level branches of each TDT.

Definition 2. *The family of all possible forms obtained per ternary S/D tree is called Ternary Inclusive Forms (TIFs).*

The numbers of these TIFs per TDT for two variables are shown on top of each S/D TDT in Figure 5. By observing Figure 5, we can generate the corresponding flattened forms by two methods. A classical method, per analogy with well-known binary forms, would be to create every transform matrix for every TIF S/D tree and then expand using that transform matrix. A better method is to create one flattened

form which is an expansion over certain transform matrix (i.e., certain TIF) and then transform systematically from one form to another form without the need to create all transform matrices from the corresponding S/D trees. This general approach can lead to several algorithms of various complexities that generalize the existing binary algorithms to obtain the corresponding forms such as FPRM, GRM and IF forms.

Example 3. *Using the result of Example 2 for the expansion of $f(x_1,x_2)$ in terms of the ternary Shannon expansion (that resembles the S/D tree for Shannon nodes in both levels):*

$$f = {}^0x_1\ {}^1x_2 + 2\cdot {}^0x_1\ {}^2x_2 + 2\cdot {}^1x_1\ {}^0x_2 + 2\cdot {}^1x_1\ {}^1x_2 + 2\cdot {}^1x_1\ {}^2x_2 + {}^2x_1\ {}^0x_2 + 2\cdot {}^2x_1\ {}^2x_2, \tag{29}$$

We can substitute any of Equations (7) - (15), or a mix of these Equations, to transform one flattened form to another. For example, if we substitute Equations (7) and (11), we obtain:

$$\begin{aligned} f =& (2(x_1)^2+1)(2(x_2')^2+x_2') + 2(2(x_1)^2+1)^2x_2 + 2(2(x_1')^2+x_1')(2(x_2)^2+1) \\ &+ 2(2(x_1')^2+x_1')(2(x_2')^2+x_2') + 2(2(x_1')^2+x_1')\cdot {}^2x_2 + {}^2x_1(2(x_2)^2+1) \\ &+ 2\cdot {}^2x_1\ {}^2x_2, \end{aligned} \tag{30}$$

By utilizing GF addition and multiplication operators from Figure 1, Equation (30) can be transformed to:

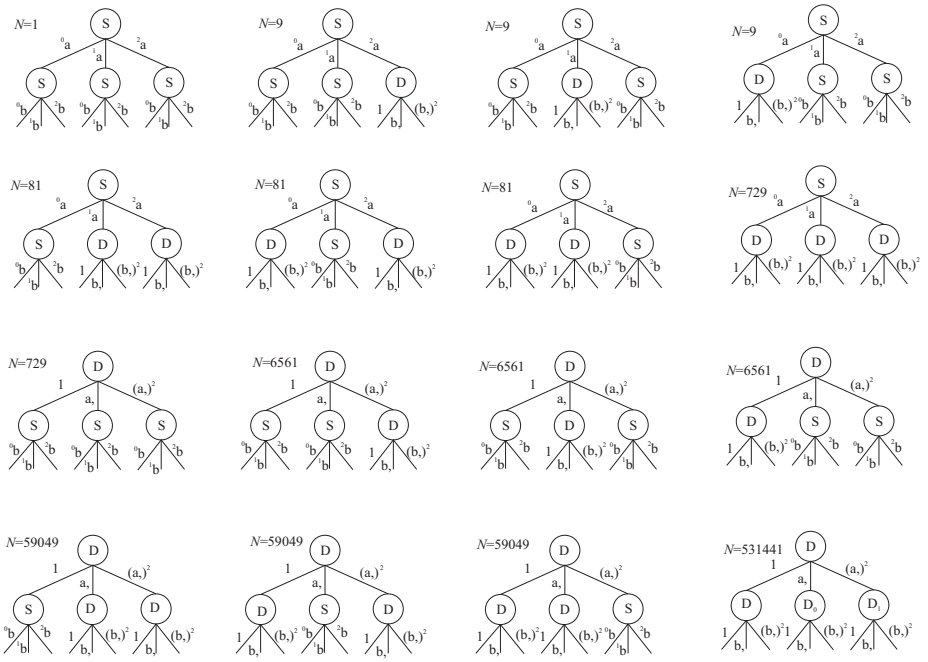
$$\begin{aligned} f =& (x_1)^2(x_2')^2 + 2(x_1)^2(x_2') + 2(x_2')^2 + x_2' + (x_1)^2({}^2x_2) + 2({}^2x_2) + 2(x_1')^2(x_2)^2 \\ &+ (x_1')^2 + (x_1')(x_2)^2 + 2x_1' + 2(x_1')^2(x_2')^2 + (x_1')^2(x_2') + (x_1')(x_2')^2 + 2x_1'x_2' \\ &+ (x_1')^2\cdot {}^2x_2 + 2(x_1')^2x_2 + 2({}^2x_1)(x_2)^2 + {}^2x_1 + 2({}^2x_1)({}^2x_2). \end{aligned} \tag{31}$$

Let us define, as one of possible definitions, the cost of the flattened form (expression) to be:

$$\text{Cost} = \#\,\text{Cubes} \tag{32}$$

Then, we observe that Equation (29) has the cost of seven, while Equation (31) has the cost of 19. Thus, the inverse transformations applied to Equation (31) would lead to Equation (29) and a reduction of cost from 19 to seven. Using the same approach, we can generate a subset of possible GFSOP expressions (flattened forms). Note that all of these GFSOP expressions are equivalent since they produce the same function in different forms. Yet, as can be observed from Equation (31), by further transformations of Equation (29) from one form to another, some transformations produce flattened forms with a smaller number of cubes than the others. From this observation rises the idea of a possible application of evolutionary computing using the S/D trees and related transformations to produce the corresponding minimum GFSOPs. Analogous to the binary case in Equation (27), the ternary GIFs can be defined as the union of ternary IFs.

Definition 3. *The family of forms, which is created as a union of sets of TIFs for all variable orders, is called Ternary Generalized Inclusive Forms (TGIFs).*



(a)

Figure 5: Ternary IF (TIF) S/D trees and their numbers: (a) 2-variable order $\{a,b\}$ and (b) 2-variable order $\{b,a\}$. Variables $\{a,b\}$ are defined as in Eq. (28) for generalized Davio (D) where in this figure $(a, \equiv \underline{a})$ and $(b, \equiv \underline{b})$.

Theorem 2. *The total number of the ternary IFs, for two variables and for orders $\{a,b\}$ and $\{b,a\}$, and the total number of ternary Generalized IFs, for two variables, are respectively:*

$$\begin{aligned} \#\mathrm{TIF}_{a,b} &= 1\cdot(3)^0+3\cdot(3)^2+3\cdot(3)^4+2\cdot(3)^6+3\cdot(3)^8+3\cdot(3)^{10}+1\cdot(3)^{12} \\ &= 730,000, \end{aligned} \tag{33}$$

$$\begin{aligned} \#\mathrm{TIF}_{b,a} &= 1\cdot(3)^0+3\cdot(3)^2+3\cdot(3)^4+2\cdot(3)^6+3\cdot(3)^8+3\cdot(3)^{10}+1\cdot(3)^{12} \\ &= 730,000, \end{aligned} \tag{34}$$

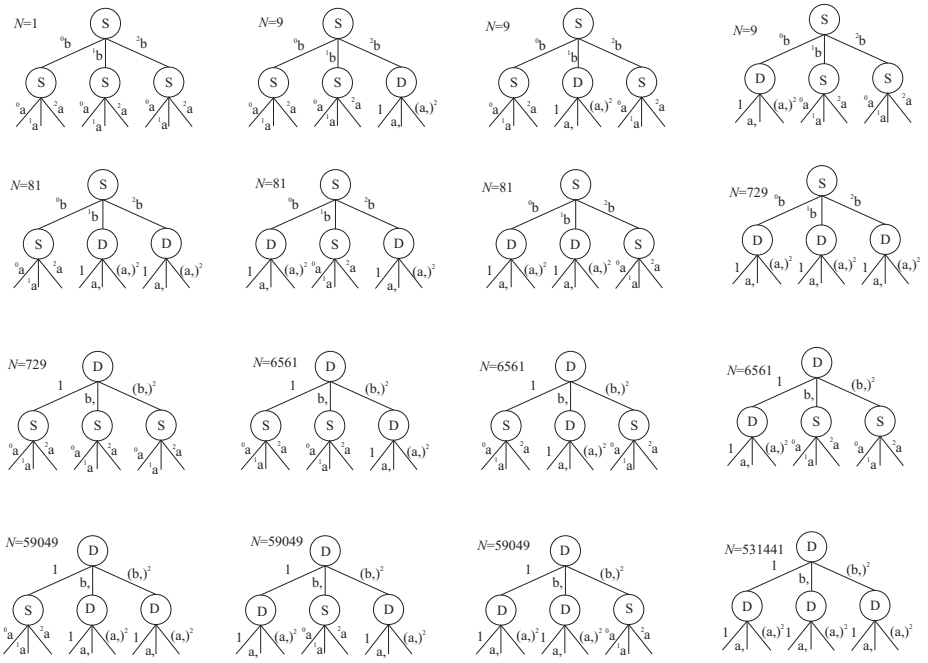
$$\begin{aligned} \#\mathrm{TGIF} &= \#\mathrm{TIF}_{a,b}+\#\mathrm{TIF}_{b,a-}\#(\mathrm{TIF}_{a,b}\cap\mathrm{TIF}_{b,a}) = 2\cdot\#\mathrm{TIF}-\#(\mathrm{TIF}_{a,b}\cap\mathrm{TIF}_{b,a}) \\ &= 2\cdot(730,000)-(1\cdot(3)^0+2\cdot(3)^6+1\cdot(3)^{12}) = 927,100. \end{aligned} \tag{35}$$

Proof. By observing Figure 5, we note that the total number of TIFs for orders $\{a,b\}$ and $\{b,a\}$ is the sum of the numbers on top of S/D trees that leads to Equations (33) - (35), respectively. □

2.2 Properties of TIFs and TGIFs

The following present basic properties of the presented TIFs and TGIFs.

Theorem 3. *Each Ternary Inclusive Form (TIF) is canonical.*



(b)

Figure 5: (continued).

Proof. An expansion is canonical iff its terms are linearly independent; none of the terms is equal to a linear combination of other terms. Using this fact, it was proven that all GF(2) IFs are canonical. Analogously over GF(3), for any function F of the same number of variables, there exists one and only one set of coefficients a_i such that F is uniquely TIF-expandable using a_i and thus the resulting expression is canonical. By induction on number of variables, terms in TIFs will be linearly independent and thus canonical. □

Theorem 4. *Ternary Generalized Inclusive Forms (TGIFs) are canonical with respect to given variable order.*

Proof. Since TGIF is the union of the corresponding TIFs, and since each TIF is canonical, then the resulting union of the corresponding canonical TIFs will be also canonical. □

For different variable orderings, some forms are not repeated while other forms are. Therefore the union of sets of TIFs for all variable orders contains more forms than any of the TIF sets taken separately and less forms than total sum of all TIFs. Generalized Inclusive Forms include other forms such as GRMs over GF(3) as can be shown by considering all possible combinations of literals for all possible orders of variables. If we relax the requirement of fixed variable ordering, and allow any ordering of variables in branches of the tree but do not allow repetitions of variables in the branches, we generate more general GF(3) family of forms.

Definition 4. *The family of forms, generated by S/D tree with no fixed ordering of variables, with variables not repeated along same branches, is called Ternary Free Generalized Inclusive Forms (TFGIFs).*

2.3 An Extended Green-Sasao Hierarchy with a New Sub-Family for Ternary Reed-Muller Logic

The Green-Sasao hierarchy of families of canonical forms [1, 10-13, 17, 24, 26] and the corresponding decision trees and diagrams is based on three generic expansions, Shannon, positive and negative Davio expansions. For example, this includes Shannon decision trees and diagrams, positive and negative Davio decision trees and diagrams, fixed polarity Reed-Muller decision trees and diagrams, Kronecker decision trees and diagrams, pseudo Reed-Muller decision trees and diagrams, pseudo Kronecker decision trees and diagrams, and linearly-independent decision trees and diagrams [1, 17, 24]. A set-theoretic relationship between families of canonical forms over GF(2) was proposed and extended by introducing binary IF, GIF and free GIF (FGIF) forms where Figure 6(a) illustrates set-theoretic relationship between the various families of canonical forms over GF(2).

Analogously to the Green-Sasao hierarchy of binary Reed-Muller families of spectral transforms over GF(2) that is shown in Figure 6(a), we will introduce the extended Green-Sasao hierarchy of spectral transforms, with a new sub-family for ternary Reed-Muller logic over GF(3). While Definitions 2 - 4 defined the Ternary Inclusive Forms, Ternary Generalized Inclusive Forms and Ternary Free Generalized Inclusive Forms, respectively, and analogously to the binary Reed-Muller case, the following definitions are introduced for the corresponding canonical expressions over GF(3).

Definition 5. *The decision tree that results from applying the ternary Shannon expansion in Equation (23) recursively to a ternary input-ternary output logic function (i.e., all levels in a DT) is called Ternary Shannon Decision Tree (TSDT). The result expression (flattened form) from the TSDT is called ternary Shannon expression.*

Definition 6. *The decision trees that result from applying the ternary Davio expansions in Equations (24) - (26) recursively to a ternary-input ternary-output logic function (i.e., all levels in a DT) are called: Ternary Zero-Polarity Davio Decision Tree (TD_0DT), Ternary First-Polarity Davio Decision Tree (TD_1DT) and Ternary Second-Polarity Davio Decision Tree (TD_2DT), respectively. The resulting expressions (flattened forms) from TD_0DT, TD_1DT and TD_2DT are called TD_0, TD_1 and TD_2 expressions, respectively.*

Definition 7. *The decision tree that results from applying any of the ternary Davio expansions (nodes) for all nodes in each level (variable) in the DT is called Ternary Reed-Muller Decision Tree (TRMDT). The corresponding expression is called Ternary Fixed Polarity Reed-Muller (TFPRM) expression.*

Definition 8. *The decision tree that results from using any of the ternary Shannon (S) or Davio (D0, D1 or D2) expansions (nodes) for all nodes in each level (variable) in the DT (that has fixed order of variables) is called Ternary Kronecker Decision Tree (TKRODT). The resulting expression is called ternary Kronecker expression.*

Definition 9. *The decision tree that results from using any of the ternary Davio expansions (nodes) for each node (per level) of the DT is called Ternary Pseudo-Reed-Muller Decision Tree (TPRMDT). The resulting expression is called ternary pseudo-Reed-Muller expression.*

Definition 10. *The decision tree that results from using any of the ternary Shannon expansion or ternary Davio expansions (nodes) for each node (per level) of the DT is called Ternary Pseudo-Kronecker Decision Tree (TPKRODT). The resulting expression is called ternary pseudo-Kronecker expression.*

Definition 11. *The decision tree that results from using any of the ternary Shannon expansion or ternary Davio expansions (nodes) for each node (per level) of the DT, disregarding order of variables, provided that variables are not repeated along the same branches, is called Ternary Free Kronecker Decision Tree (TFKRODT). The result is called ternary free-Kronecker expression.*

Definition 12. *The ternary Kronecker DT that has at least one ternary generalized Reed-Muller expansion node is called Ternary Generalized Kronecker Decision Tree (TGKDT). The result is called ternary generalized Kronecker expression.*

Definition 13. *The ternary Kronecker DT that has at least one TGIF node is called Ternary Generalized Inclusive Forms Kronecker (TGIFK) Decision Tree. The result is called ternary generalized Inclusive Form Kronecker expression.*

Figure 6(b) illustrates the extended GF(3) Green-Sasao hierarchy with the new sub-family (TGIFK). The presented TGIF nodes can be realized with Universal Logic Modules (ULMs) for pairs of variables, as will be shown in Section 3, which is analogous to what was done for the binary case. Although the S/D trees that have been developed so far are for the ternary radix, similar S/D trees can be developed as well for higher Galois radices of GF(p^k) where p is a prime number and k is a natural number ≥ 1.

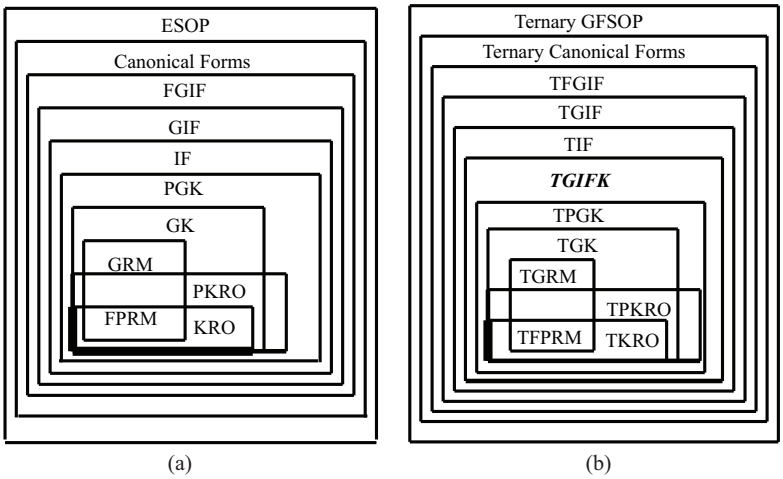


Figure 6: Green-Sasao hierarchy: (a) set-theoretic relationship between GF(2) families of canonical forms, and (b) the extended Green-Sasao hierarchy with a new sub-family (TGIFK) for GF(3) Reed-Muller logic.

3 Universal Logic Modules For The Circuit Realization Of S/D Trees

The nonsingular expansions of ternary Shannon (S) and ternary Davio (D0, D1 and D2), can be realized using a Universal Logic Module (ULM) with control variables corresponding to the variables of the basis functions which are the variables we are expanding upon. We call it a Universal Logic Module, because similarly to a multiplexer, all functions of two variables can be realized with two-level trees of such modules using constants on the second-level data inputs.

The presented ULMs are complete systems because they can implement all possible functions with certain number of variables. The concept of the universal logic module was used for binary RM logic over GF(2), as well as the general case of linearly independent (LI) logic that includes RM logic as a special case. Binary LI logic extended the universal logic module from just being a multiplexer (Shannon expansion), AND/EXOR gate (positive Davio expansion) and AND/EXOR/NOT gate with inverted control variable (negative Davio expansion), to the universal logic modules for any expansion over any linearly independent basis functions. Analogously to the binary case, Figure 7 presents universal logic modules for ternary Shannon and ternary Davio, respectively. One can note, that any function f can be produced by the application of the independent variable x and the cofactors $\{f_i, f_j$ and $f_k\}$ as inputs to a ULM. The form of the resulting function depends on our choice of the shift and power operations that we choose inside the ULM for the input independent variable, and on our choice of the weighted combinations of the input cofactors. Utilizing this note, we can combine all Davio ULMs to create the single all-Davio ULM, where Figure 7(c) illustrates this ULM. Also, the more general ULM as shown in Figure 7(d), can be generated to implement all ternary GF(3) Shannon and Davio expansions.

In general, the gates in the ULMs can be implemented, among other circuit technologies, by using binary logic over GF(2) or using multiple-valued circuit gates. Each ternary ULM corresponds to a single node in the nodes of ternary DTs that were illustrated previously. The main advantage of such powerful ULMs is in high layout regularity that is required by future nanotechnologies, where the trees can be realized in efficient layout because they do not grow exponentially for practical functions. For instance, assuming ULM from Figure 7, although every two-variable function can be realized with four such modules, it is highly probable that most of two-variable functions will require less than four modules. Because of these properties, this approach is further expected to give good results when applied to the corresponding incompletely specified functions and multiple-valued relations.

4 Conclusions and Future Work

In this paper, an extended Green-Sasao hierarchy of families and forms is introduced. Analogously to the binary case, two general families of canonical ternary Reed-Muller forms, called Ternary Inclusive Forms (TIFs) and their generalization of Ternary Generalized Inclusive Forms (TGIFs), are presented. The second family includes minimum GF(3) Galois Field Sum-Of-Products (GFSOPs). The multiple-valued Shannon-Davio (S/D) trees developed in this paper provide more general polarity with regards to the

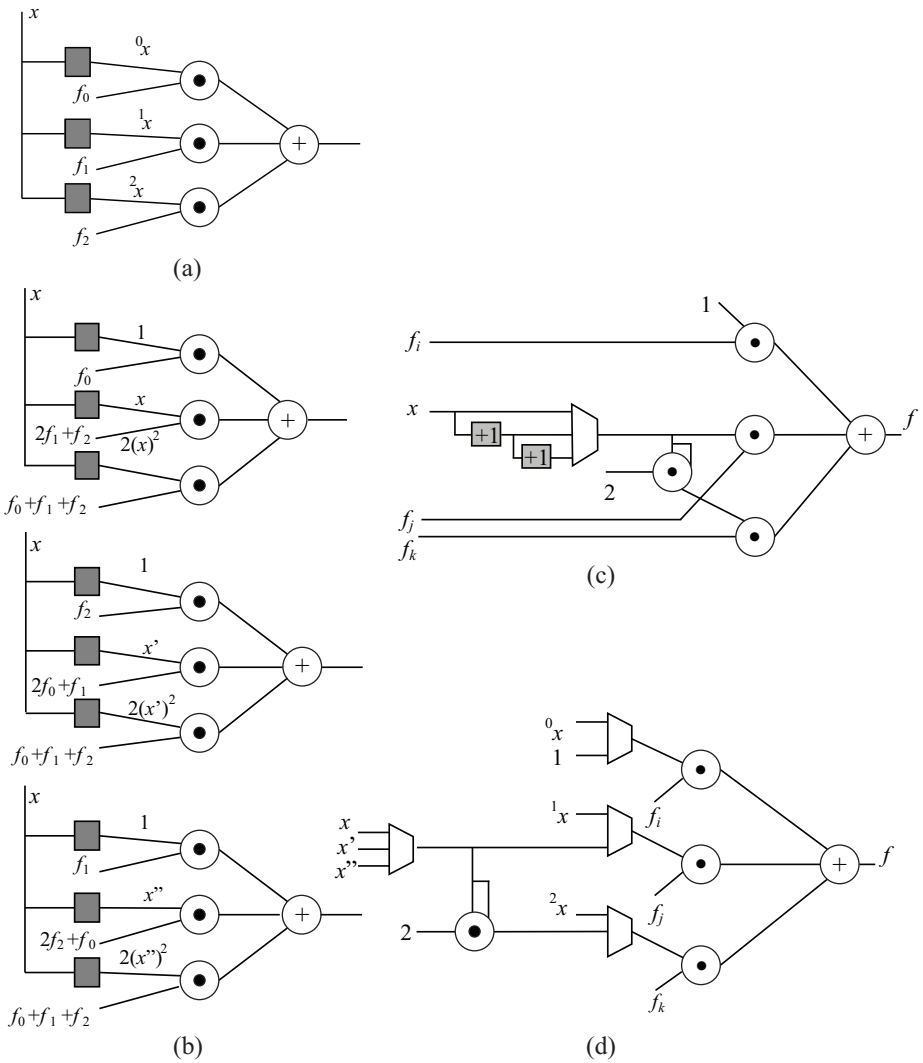


Figure 7: Various ternary ULMs: (a) ULM of GF(3) Shannon using three GF(3) multiplication gates and one GF(3) addition gate, (b) ULMs of GF(3) Davio $\{D_0, D_1, D_2\}$ using three GF(3) multiplication gates and one GF(3) addition gate, (c) ULM for all GF(3) Davio expansions using four GF(3) multiplication gates, one GF(3) addition gate, two shift operators, and one multiplexer, and (d) general ULM of GF(3) S/D node using four GF(3) multiplication gates, one GF(3) addition gate and four multiplexers.

corresponding IF polarity, which contains the GRM as a special case. Since Universal Logic Modules (ULMs) are complete systems that can implement all possible logic functions utilizing S/D expansions of multiple-valued Shannon and Davio decompositions, the realization of the presented S/D trees utilizing the corresponding ULMs is also introduced. The application of the presented TIFs and TGIFs can be used to find minimum GFSOP for multiple-valued inputs-outputs within logic synthesis, where a GFSOP minimizer based on IF polarity can be used to minimize the corresponding multiple-valued GFSOP expressions.

Future work will include the following items:

1. The investigation of the various multiple-valued and two-valued techniques for the ULM implementation for the important case of quaternary logic. Since the ULM realization over GF(3) extends and generalizes from the binary case of the GF(2) ULM realization - where most digital system designs are performed - the special case of the extension into GF(4) becomes important as this GF(4) ULM realization can be achieved by utilizing the currently existing and widely-used implementations over GF(2). This two-valued realization of quaternary ULM can be done for GF(4) addition utilizing GF(2) addition using vector of EXORs, (4/2) and (2/4) decoders, and for GF(4) multiplication utilizing GF(2) operations using vectors of XORs, ANDs, (4/2) and (2/4) decoders, and therefore the GF(4) ULM producing quaternary Shannon and all Davio expansions can be achieved using the corresponding multiplexers and GF(4) additions and multiplications which can be accordingly realized using GF(2) methods.

2. The implementation of the introduced ULMs using nanotechnology methods such as quantum computing, quantum dots and carbon nanotubes.

3. The utilization of evolutionary algorithms for function minimization using the introduced S/D trees.

4. The investigation using other more complex types of literals such as the presented generalized literal (GL) and universal literal (UL) to expand upon and consequently construct the corresponding new ULMs.

Acknowledgment

This research was performed during sabbatical leave in 2015-2016 granted from The University of Jordan and spent at Philadelphia University.

References

[1] A. N. Al-Rabadi, Reversible Logic Synthesis: From Fundamentals to Quantum Computing, Springer-Verlag, 2004.

[2] A. N. Al-Rabadi, "Three-Dimensional Lattice Logic Circuits, Part III: Solving 3D Volume Congestion Problem," Facta Universitatis - Electronics and Energetics, Vol. 18, No. 1, pp. 29 - 43, 2005.

[3] A. N. Al-Rabadi, "Three-Dimensional Lattice Logic Circuits, Part II: Formal Methods," Facta Universitatis - Electronics and Energetics, Vol. 18, No. 1, pp. 15 - 28, 2005.

[4] A. N. Al-Rabadi, "Three-Dimensional Lattice Logic Circuits, Part I: Fundamentals," Facta Universitatis - Electronics and Energetics, Vol. 18, No. 1, pp. 1 - 13, 2005.

[5] A. N. Al-Rabadi, "Quantum Logic Circuit Design of Many-Valued Galois Reversible Expansions and Fast Transforms," Journal of Circuits, Systems, and Computers, World Scientific, Vol. 16, No. 5, pp. 641 - 671, 2007.

[6] A. N. Al-Rabadi, "Representations, Operations, and Applications of Switching Circuits in the Reversible and Quantum Spaces," Facta Universitatis - Electronics and Energetics, Vol. 20, No. 3, pp. 507 - 539, 2007.

[7] A. N. Al-Rabadi, "Multi-Valued Galois Shannon-Davio Trees and their Complexity," Facta Universitatis - Electronics and Energetics, Vol. 29, No. 4, pp. 701-720, 2016.

[8] B. Falkowski and L.-S. Lim, "Gray Scale Image Compression Based on Multiple-Valued Input Binary Functions, Walsh and Reed-Muller Spectra," Proc. ISMVL, 2000, pp. 279-284.

[9] H. Fujiwara, Logic Testing and Design for Testability, MIT Press, 1985.

[10] D. H. Green, "Families of Reed-Muller Canonical Forms," Int. J. Electronics, No. 2, pp. 259-280, 1991.

[11] M. Helliwell and M. A. Perkowski, "A Fast Algorithm to Minimize Multi-Output Mixed-Polarity Generalized Reed-Muller Forms," Proc. Design Automation Conference, 1988, pp. 427-432.

[12] S. L. Hurst, D. M. Miller, and J. C. Muzio, Spectral Techniques in Digital Logic, Academic Press Inc., 1985.

[13] M. G. Karpovsky, Finite Orthogonal Series in the Design of Digital Devices, Wiley, 1976.

[14] S. M. Reddy, "Easily Testable Realizations of Logic Functions," IEEE Trans. Comp., C-21, pp. 1183-1188, 1972.

[15] T. Sasao and M. Fujita (editors), Representations of Discrete Functions, Kluwer Academic Publishers, 1996.

[16] T. Sasao, "Easily Testable Realizations for Generalized Reed-Muller Expressions," IEEE Trans. Comp., Vol. 46, pp. 709-716, 1997.

[17] R. S. Stanković, Spectral Transform Decision Diagrams in Simple Questions and Simple Answers, Nauka, 1998.

[18] R. S. Stanković, C. Moraga, and J. T. Astola, "Reed-Muller Expressions in the Previous Decade," Proc. Reed-Muller, Starkville, 2001, pp. 7-26.

[19] S. B. Akers, "Binary Decision Diagrams," IEEE Trans. Comp., Vol. C-27, No. 6, pp. 509-516, June 1978.

[20] R. E. Bryant, "Graph-based Algorithms for Boolean Functions Manipulation," IEEE Trans. on Comp., Vol. C-35, No.8, pp. 667-691, 1986.

[21] D. K. Pradhan, "Universal Test Sets for Multiple Fault Detection in AND-EXOR Arrays," IEEE Trans. Comp., Vol. 27, pp. 181-187, 1978.

[22] M. Cohn, Switching Function Canonical Form over Integer Fields, Ph.D. Dissertation, Harvard University, 1960.

[23] R. Drechsler, A. Sarabi, M. Theobald, B. Becker, and M. A. Perkowski, "Efficient Representation and Manipulation of Switching Functions Based on Ordered Kronecker Functional Decision Diagrams," Proc. DAC, 1994, pp. 415-419.

[24] B. J. Falkowski and S. Rahardja, "Classification and Properties of Fast Linearly Independent Logic Transformations," IEEE Trans. on Circuits and Systems-II, Vol. 44, No. 8, pp. 646-655, August 1997.

[25] A. Gaidukov, "Algorithm to Derive Minimum ESOP for 6-Variable Function," Proc. Int. Workshop on Boolean Problems, Freiberg, 2002, pp. 141-148.

[26] S. Hassoun, T. Sasao, and R. Brayton (editors), Logic Synthesis and Verification, Kluwer Acad. Publishers, 2001.

[27] C. Y. Lee, "Representation of Switching Circuits by Binary Decision Diagrams," Bell Syst. Tech. J., Vol. 38, pp. 985-999, 1959.

[28] C. Moraga, "Ternary Spectral Logic," Proc. ISMVL, pp. 7-12, 1977.

[29] J. C. Muzio and T. Wesselkamper, Multiple-Valued Switching Theory, Adam-Hilger, 1985.

[30] R. S. Stanković, "Functional Decision Diagrams for Multiple-Valued Functions," Proc. ISMVL, 1995, pp. 284-289.

[31] B. Steinbach and A. Mishchenko, "A New Approach to Exact ESOP Minimization," Proc. Reed-Muller, Starkville, 2001, pp. 66-81.

[32] I. Zhegalkin, "Arithmetic Representations for Symbolic Logic," Math. Sb., Vol. 35, pp. 311-377, 1928.

[33] A. N. Al-Rabadi, "Carbon Nano Tube (CNT) Multiplexers for Multiple-Valued Computing," Facta Universitatis - Electronics and Energetics, Vol. 20, No. 2, pp. 175 - 186, 2007.

[34] A. N. Al-Rabadi, "Closed-System Quantum Logic Network Implementation of the Viterbi Algorithm," Facta Universitatis - Electronics and Energetics, Vol. 22, No. 1, pp. 1 - 33, 2009.